*Real-Time Imaging*

# Gesture-Tracking in Real Time with Dynamic Regional Range Computation

Leonid V. Tsap

Center for Applied Scientific Computing
University of California Lawrence Livermore National Laboratory
P. O. Box 808, L-551, Livermore, CA 94551
e-mail: tsap1@llnl.gov

## Abstract[1]

This paper presents a new approach to the range data utilization in a gesture-tracking system. Using three-dimensional data is essential for human motion analysis; however, the speed of complete range estimation prohibits from including it in most real-time systems. This work describes a gesture-tracking system using real-time local range on-demand. The system represents a gesture-controlled interface for interactive visual exploration of large data sets. The paper describes a method performing range processing only when necessary and where necessary. Range data is processed only for non-static regions of interest. This is accomplished by a set of filters on the color, motion, and range data. The speedup achieved is between 1.70 and 2.15. The algorithm also includes a robust skin-color segmentation insensitive to illumination changes. Selective range processing results in dynamic regional range images that contain only information needed by the system.

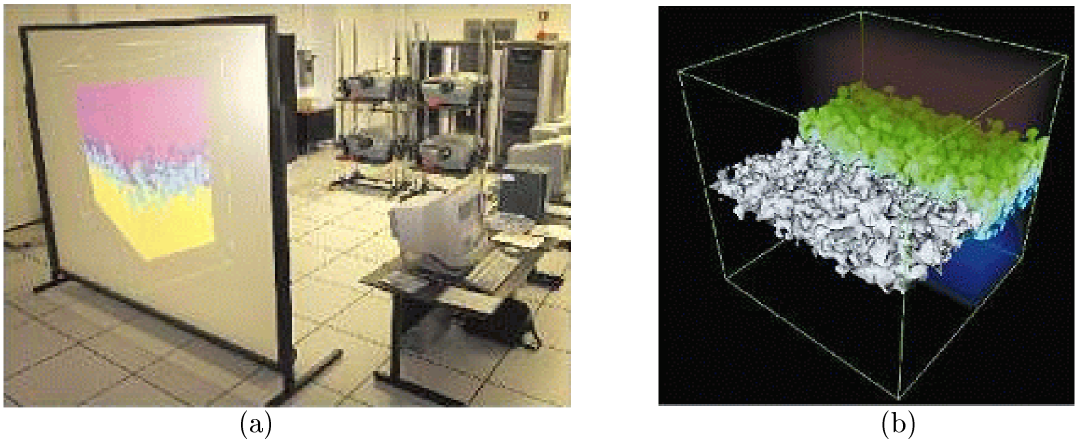
[1] This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract number W-7405-Eng-48. UCRL-JC-137651-Rev1

# Contents

# 1 Introduction

## 1.1 Applicational Requirements

Recent years have seen a drastic increase in the size and complexity of scientific data. The National Institutes of Health's (NIH) Visible Human project generated data sets of a single 3-D volume consisting of 12 billion elements. Nearly a terabyte of satellite data is produced daily. Advanced physics simulation here, at Lawrence Livermore National Laboratory (LLNL), is responsible for generating large data sets, which are expected to increase to one terabyte every five minutes by 2004. Traditional visualization represents the last step in data processing. However, the efficiency of such processing suffers when errors are discovered at this point, and the entire data analysis cycle has to start over. Therefore, the trend in data growth is amplified by increasing requirements for interactive data access, display, exploration, analysis and collaboration. Focused on the development of efficient techniques addressing these requirements, the SAVAnTS (Scalable Algorithms for Visualization and Analysis of Terascale Science) project is a collaboration between the Center for Applied Scientific Computing at LLNL and multiple academic partners. With such substantial amounts of data to explore, we are also interested in developing new interactive settings that would allow scientists to explore their data in a more intuitive environment. The data would be projected onto a large screen (Figure 1(a)), and updated in real-time following gesture-based commands of interacting scientists. The gesture-tracking system described in this paper will be responsible for supplying data manipulation parameters to interactive data exploration and collaborative visualization software (Figure 1(b)), and to virtual reality systems.



(a) (b)

Figure 1: (a) A four-projector display with a 2560x2048 resolution used for interactive applications. (b) An example of intaractive manipulation of isosurface and volume rendering parameters.

Since the system is being developed as a front end for gesture-controlled large-scale visualization and virtual reality manipulation, certain requirements and complications are obvious. First, 3-D information is required, not necessarily at a video-frame rate, but at least a few times per second (optimal parameters should be determined as a result of testing on a large group of people). Second, not only arms or hands, but also the entire body of the interacting person is moving. More over, interaction will take place in front of the large screen where the data being manipulated will be displayed. Most of the time the data will be updated dynamically as a result of such manipulations; and, therefore, traditional techniques such as background subtraction cannot easily separate a figure from the background. Third, motion of the interacting person should be natural and should result in intuitive data manipulation, where intuitive means easily learned and fast to provide immediate results.

## 1.2 Previous Work in Gesture Tracking

Object tracking from image sequences is a very important research domain. Goals of object tracking include segmenting each frame into differently moving objects, selecting the object of interest, and analyzing its motion during the entire sequence or multiple sequences. Object tracking, therefore, involves processing of both spatial and temporal data. A number of applications is dealing with tracking the motion of the human body. These applications include video-surveillance, gesture-based interfaces for multimedia applications and systems, and interfaces for people with disabilities that prevent them from using the standard input technology, and videoconferencing. The most popular mode of human-computer interaction (HCI) is based on devices like keyboards and mice, which limit the speed and naturalness of the interaction [1]. Researchers continue to investigate ways to use human communication through movement as a natural means of interacting with computers. They strive to design and develop computer interfaces that capture and interpret such human movement. Another application is object manipulation in virtual environments.

Traditional approaches to tracking typically relied on segmentation of the intensity data, using motion or appearance data. A majority of the methods began by segmenting the human body from the background. For instance, in “blob approaches” people were modeled as a number of blobs resulting from pixel classification based on their color and position in the image. Wren *et al.* [2] achieved segmentation by classifying pixels into one of several models, including a static world and a dynamic user represented by gaussian blobs. Yang and Ahuja [3] used skin color and the geometry of palm and face regions for segmentation stages of their system. A Gaussian mixture (with parameters estimated by an EM algorithm) modeled the distribution of skin-color pixels. Rehg and Kanade [4] used a 3-D hand model to track a hand. They compared line features from the images with the projected model, and performed incremental state corrections. Similar work was presented by Kuch and Huang [5] in which the synthesis process could fit the hand model to any person's hand. Bobick and Wilson [6] treated gesture as a sequence of states and computed configuration states along prototype gestures. Yacoob and Black proposed parameterized representation of human movement [7]. Cutler and Davis [8] segmented the motion and computed a moving objects self-similarity (including human motion experiments). Significant work is being performed in the area of recognition where hidden Markov models are often employed successfully ([9, 10]) by allowing researchers to address the highly stochastic nature of human gestures. A review by Aggarwal and Cai [11] classified approaches to human motion analysis, the tasks involved, and major areas related to human motion interpretation. A review by Pavlovic *et al.* [1] addressed main components and directions in gesture recognition research for HCI.

It is known that color-based skin detection techniques are susceptible to variability in lighting conditions [1]. Some common solutions included [1]: specially colored gloves or markers, restrictive backgrounds or clothing, prior knowledge of initial hand positions, or movement restrictions. A goal of our project is to exclude such simplifications. Instead, we use the SCT/Center algorithm that can handle changing illumination. It was originally developed for skin cancer detection using color features [12]. Later the algorithm was successfully tested for position estimation of micro-rovers [13]. Other similar solutions for decoupling intensity from color information were also recently investigated ([9]).

### 1.3 Motivation for Range On-Demand Approach

Usefulness of 3-D data in gesture-analysis applications is not questionable because the projection of human movement is always affected by the observation viewpoint and the distance from the camera [7]. Since machine vision systems try to recover useful information about a scene from its projections, having three-dimensional (3-D) data eliminates ambiguities in solving the inversion of a many-to-one mapping [14]. Most gesture-tracking and recognition applications could certainly benefit from including range data and having more information recovered from a scene. Until recently, however, using range data for tracking was not feasible because of the speed and cost considerations.

Some authors used multiple cameras and models to obtain 3-D locations of body parts. Azarbayejani and Pentland [15] triangulated on blobs composing a model. Gavrila and Davis [16] addressed whole-body tracking with four cameras placed in the corners of the room. Segen and Kumar [17] used depth cues from projections of the hand and its shadow for 3-D hand pose estimation. Moeslund and Granum estimated 3-D poses from a monocular system by modeling degrees of freedom and by using the analysis-by synthesis approach ([18]). Davis and Shah presented a tracking method by fitting 3-D models (generalized cylinders) to fingers in a 2-D image [14]. Otherwise, range data was used in motion analysis primarily in an offline mode [19, 20].

Recent availability of less expensive, faster range data makes it a feasible additional source of information for tracking. This is the first real-time gesture-tracking system that utilizes on-demand range in both spatial and temporal representations (some initial results have previously appeared in [21]). It will be applied to natural navigation and visualization of large data sets. The method is also applicable to virtual reality systems. Oda *et al.* [22] reported application of a real-time range to virtual reality which utilized comparison of the depth information in real and synthetic data. In addition to efficient range processing, the proposed method also deals with the major shortcoming of color-based localization methodologies variability of the skin-color classification results under different illumination conditions.

## 2 Description of the Method

The entire algorithm of the range on-demand approach is shown in Figure 2. Both color and range image are *grabbed synchronously* (box 1), and color image is extracted and rectified (corrected for lens distortions). However, *range is not processed* at this point (box 2) as one would expect. Instead, numerous filters are applied to the data as described below. These filters achieve a goal of localizing regions of interest (ROIs), specifically hands for our application (since their motion will provide input to visualization programs).

First, color feature filters are applied (box 3). The spherical coordinate transform (SCT) separates color and brightness information. Color normalization provides SCTs insensitivity to variations in illumination. LAB space is computed, and pixels are classified as skin are computed using derived statistical data. A skin classifier with minimum distance classifier using Mahalanobis distance selects pixels that can be considered skin pixels.

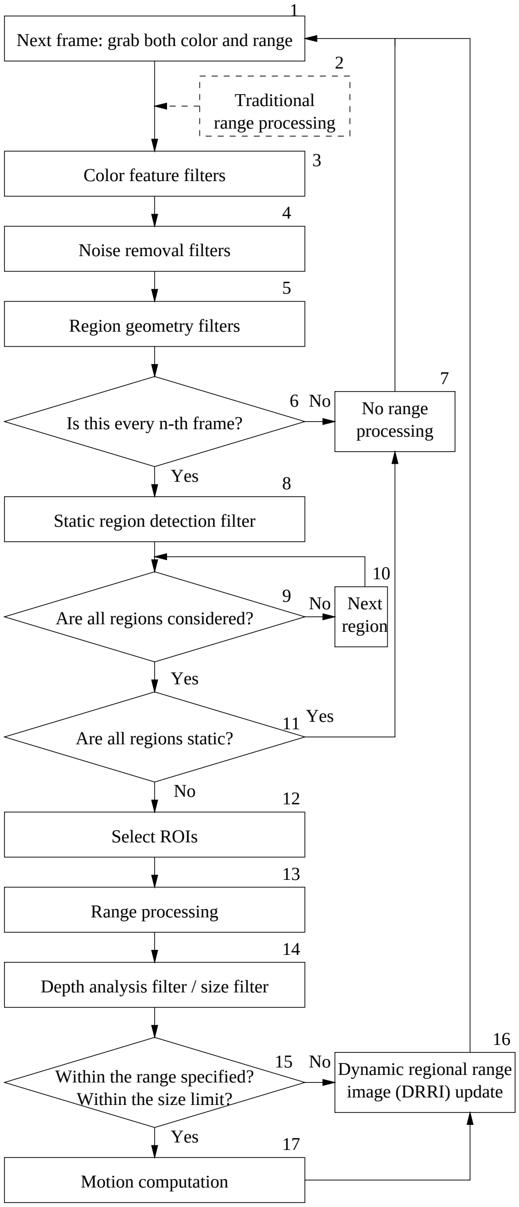

1
Next frame: grab both color and range
2
Traditional range processing
3
Color feature filters
4
Noise removal filters
5
Region geometry filters
6
Is this every n-th frame?
No
7
No range processing
Yes
8
Static region detection filter
9
Are all regions considered?
No
10
Next region
Yes
11
Are all regions static?
Yes
No
12
Select ROIs
13
Range processing
14
Depth analysis filter / size filter
15
Within the range specified? Within the size limit?
No
16
Dynamic regional range image (DRRI) update
Yes
17
Motion computation

Figure 2: Algorithm of the range on-demand approach.

Noise is removed by a sequence of erosions and dilations (box 4). The connected component analysis is performed next by scanning from left to right and from top to bottom, labeling, and evaluating equivalencies. Resulting regions are sorted, and small regions are removed from further consideration. Region area is evaluated with respect to the image size (box 5). Other geometry and shape filters are designed to eliminate regions with unlikely shapes for human faces or hands, including long regions, regions with very few pixels (less than 30 %) classified as skin.

Human hands and faces are difficult to detect when only color information is used. Experiments described in the next section use a simulation of a possible virtual scene with objects of various shapes and sizes, and a robotic hand that could be following human gestures in a completed back-end virtual reality application. To confuse the program, one of the objects, a ball in the center of the scene, is given color properties very similar to human skin.

That is why additional filters are set to prevent such confusion. A static region detection filter (defined for frames after the first) determines the absence of current motion for a given region (box 8). The filter evaluates current motion proportionally to the average noise level and the image size (since motion considered neglectable for relatively large images can be considered important for smaller ones). The process results in dynamic regional range images (DRRIs). Static regions are shown on DRRIs as outlines only, since range is not computed for them. DRRIs contain range information for regions of interest (with pixels still classified as skin after color- and geometry-based filters) moving in the current frame, outlines for static regions and recent motion information for both.

Only non-static regions (box 12) are selected for range processing (box 13), which takes place at this point (again let us note that color and range were grabbed synchronously, only the range processing was postponed). Stereo is estimated only for selected ROIs. Thus, the computation bottleneck is greatly reduced (see the next section for speedup percentages vs. region sizes).

Next, the depth analysis filter (box 14) evaluates whether ROIs exhibit face or hands geometry; since the depth is known at this step, absolute sizes are computed. Non-human regions are excluded from the motion computation, but are currently still tracked on the color images and DRRIs for visual purposes. If skin-colored moving objects pass previous filters, then they are unveiled at this point (and not included into motion computation in box 17). To increase the speed even further, range can be selected for region analysis only for every $n$ color images in box 6 (depending on the application, in this work $n$=4).

## 3 Experimental Results

The following experiments involve application of the algorithm to color and range image sequences of gestures. Triclops color stereo vision system (manufactured by Point Grey Research, Vancouver, Canada) is used to capture these sequences [24]. The module connects to a single-processor Pentium III PC.

Range information is recovered in real time from a correlation-based trinocular stereo algorithm.

Typical color and range images produced by the stereo vision system are shown in Figure 3. Closer objects appear lighter in the range data, except for the darkest areas (for instance, some hair and far wall regions), indicating that no correspondence was found during the stereo matching process. As a result of applying color information-based filters, skin regions are selected (shown as white areas in the binary image in Figure 4(a), and as rectangular enclosing boxes in Figure 4(b)).

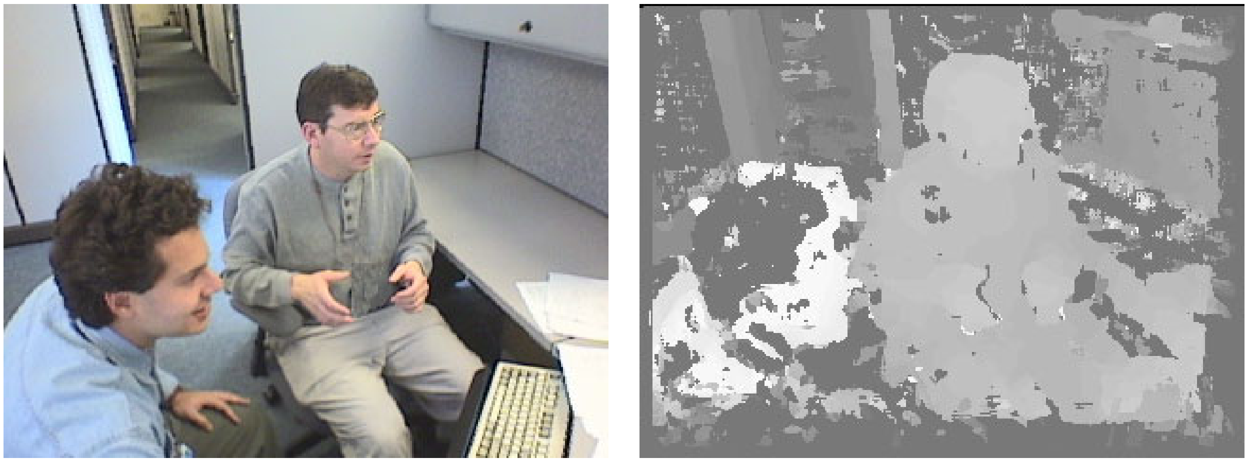
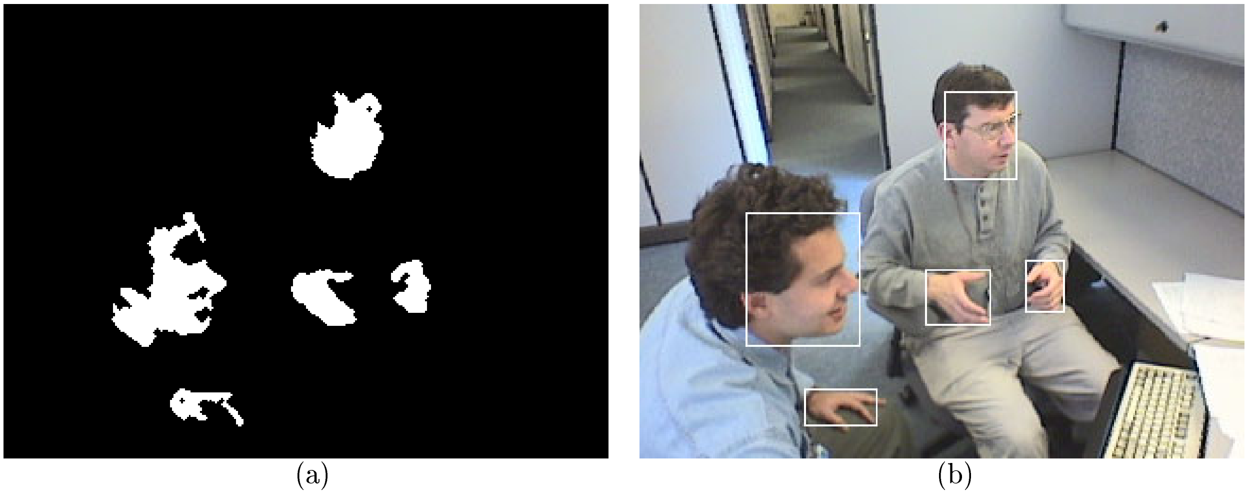


Figure 3: Typical color and range images produced by the stereo vision system.



(a)

(b)

Figure 4: Pixels classified as skin as a result of applying color information-based filters.

Selected frames from sequences of intensity images and DRRIs are shown in Figures 5 and 6. Frames are chosen when both color and range information were scheduled to be processed (every $n$-th frame in the algorithm, where $n$=5). To tell the preparation stage from the nucleus and retraction stages, interactors will use a fist as if grabbing the object being manipulated. That is why frames taken during gestures signifying object operations show persons using closed fists. Since manipulation of virtual objects is one of popular applications of hand gestures [1], the background represents a scene with virtual objects and a robotic arm-manipulator, one of them (a ball) is skin-colored. Of course, the main application of the system, as discussed in the Introduction, will be interactive exploration of visualized large data sets.

The first sequence (Figure 5) shows tracking of a zoom gesture (hand moving toward the camera). First frame processing results show that all three candidate skin regions are detected: the face, the hand, and the virtual ball (created as a distracter with color similar to skin). These regions pass color feature filters, noise removal and region geometry filters, and static region detection filters (since relative motion is not defined for the first frame). However, the ball is not tracked beyond the first frame since it is obviously a static object. Note that the tracking system developed is a front-end for the interactive visualization software, and, therefore, background subtraction is not the best option in the general case. No apparently static regions are processed (the face and the ball). Otherwise, they would have been excluded from motion analysis on the basis of depth or size or both by the last filters. Hand motion was detected in frames 3, 7 and 10, and reflected in respective DRRIs.

Similarly, the second sequence tracking a translational gesture (hand motion side-to-side in Figure 6) contains hand motion in six frames (not considering the first one) and head motion in two frames (8 and 9). Obviously, keeping one's head completely motionless is not a practical consideration, and head motion is present in all frames. In most frames, however, it does not pass the small motion filter (based on the average noise level and the image size). The number of frames when the range is computed is also a function of the motion velocity (greater in the second sequence which results in more inter-frame hand motion passing the "static" filter).

DRRIs can be included in motion analysis, trajectory computation, gesture recognition, to determine what types of gestures are natural and feasible for robust tracking and interpretation for interactive exploration of large data sets and virtual environments. They can be plotted in a 3-D space for movement trajectory parameterization. Also they can produce (also in 3-D) templates for recognition of movements somewhat similar to the temporal templates [23].

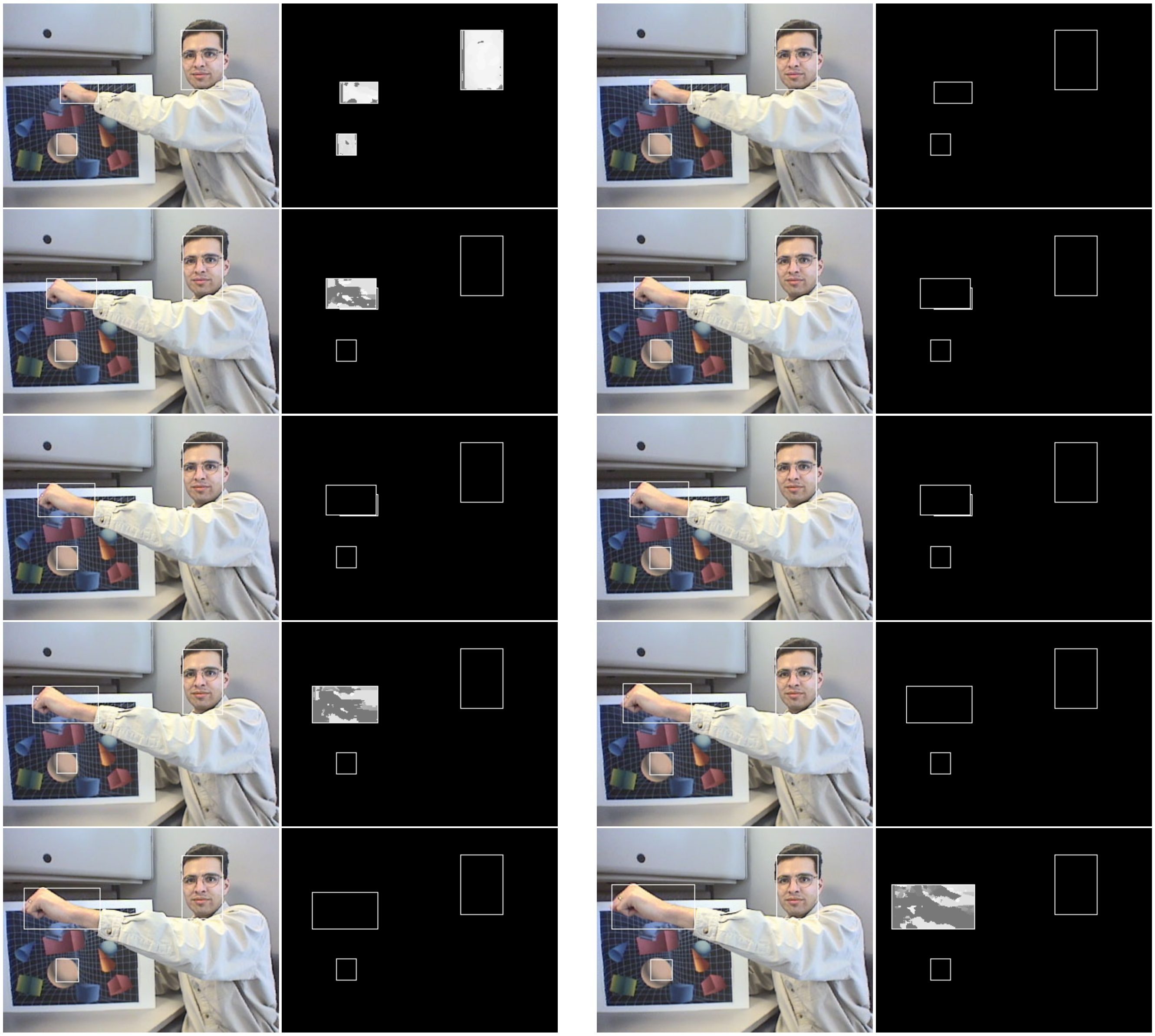


Figure 5: Tracking of skin-color regions (first and third columns) and progress in DRRIs (second and fourth columns) for zoom gesture (hand moving towards the camera).

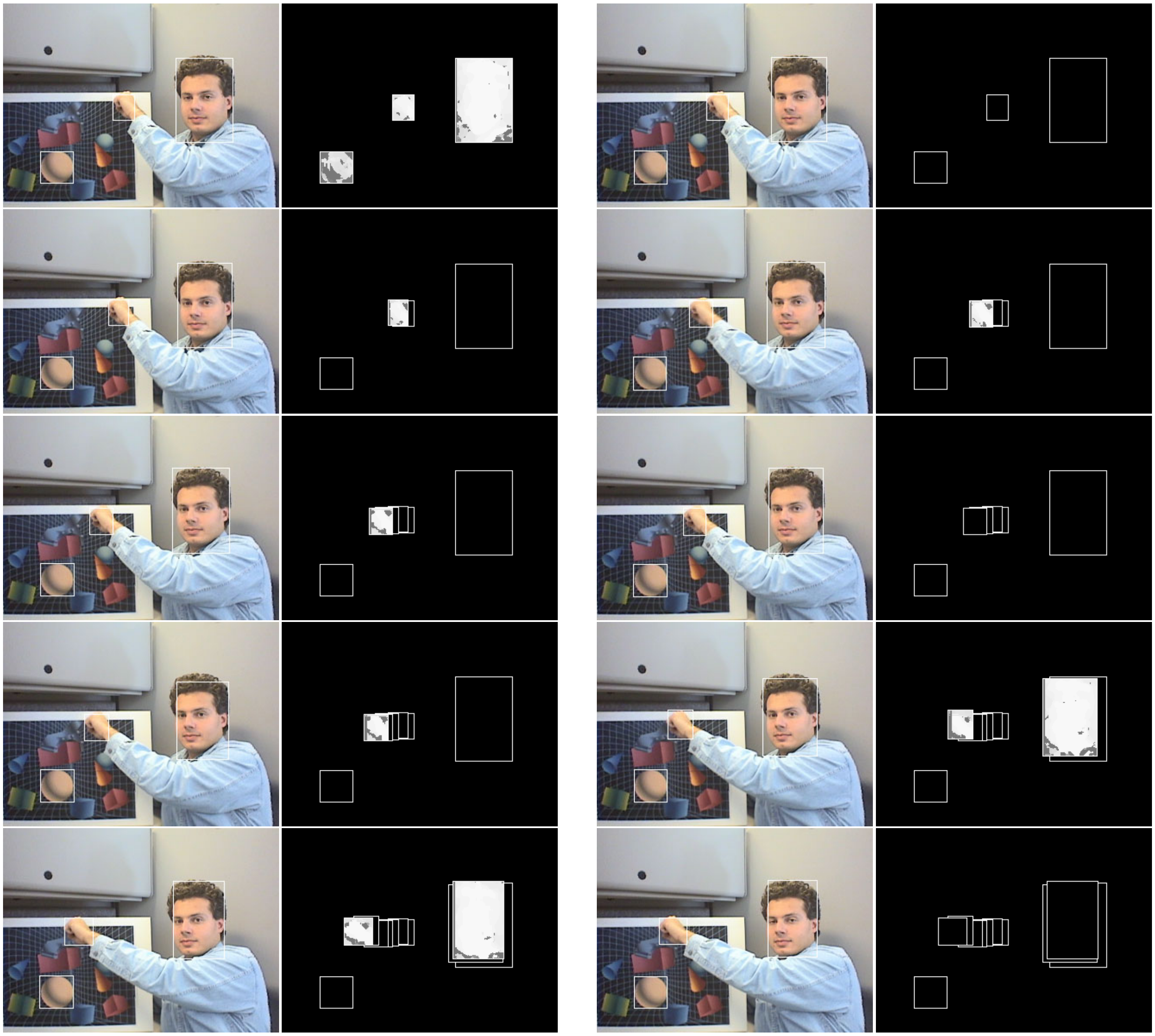


Figure 6: Tracking of skin-color regions (first and third columns) and progress in DRRIs (second and fourth columns) for translation gesture (hand moving side-to-side).

Tables 1 and  2 contain statistics for corresponding motion sequences. Frame numbers correspond to respective frames in figures. “Number of ROIs” column indicates regions selected for range processing, next column contains their total area. Percentage of total image size is also included, as well as the total time for this frame (for the entire algorithm to process it). The speedup is defined as a ratio of an average non-ROI processing time per frame (488 ms) to the actual running time for this frame (with the new method). It is defined for all possible types of motion under any circumstances. Even though the second movement triggers more frames where range computation is needed (due to the greater motion velocity), very few (necessary) regions are selected for processing.

Table 1: Statistics for the forward hand motion.

| Frame (Fig. 5) | Number of ROIs | Total area of ROIs | % of area size | Time for this frame, ms | Speedup |
|---|---|---|---|---|---|
| 1 | 3 | 5105 | 6.65 | 281 | 1.74 |
| 2 | 0 | 0 | 0 | 240 | 2.03 |
| 3 | 1 | 2030 | 2.64 | 261 | 1.87 |
| 4 | 0 | 0 | 0 | 233 | 2.09 |
| 5 | 0 | 0 | 0 | 227 | 2.15 |
| 6 | 0 | 0 | 0 | 233 | 2.09 |
| 7 | 1 | 3268 | 4.26 | 274 | 1.78 |
| 8 | 0 | 0 | 0 | 233 | 2.09 |
| 9 | 0 | 0 | 0 | 234 | 2.09 |
| 10 | 1 | 4992 | 6.50 | 280 | 1.74 |

Table 2: Statistics for the side-to-side hand motion.

| Frame (Fig. 6) | Number of ROIs | Total area of ROIs | % of area size | Time for this frame, ms | Speedup |
|---|---|---|---|---|---|
| 1 | 3 | 8690 | 11.32 | 287 | 1.70 |
| 2 | 0 | 0 | 0 | 240 | 2.03 |
| 3 | 1 | 690 | 0.90 | 267 | 1.83 |
| 4 | 1 | 837 | 1.09 | 267 | 1.83 |
| 5 | 1 | 837 | 1.09 | 267 | 1.83 |
| 6 | 0 | 0 | 0 | 233 | 2.09 |
| 7 | 1 | 868 | 1.13 | 267 | 1.83 |
| 8 | 2 | 6719 | 8.75 | 281 | 1.74 |
| 9 | 2 | 6425 | 8.37 | 280 | 1.74 |
| 10 | 0 | 0 | 0 | 241 | 2.02 |

Average frame rate for longer sequences is also measured and averaged. Processing 1 range image for every 4 color images is done at a rate of 10.6 frames per second for a 320x240 image size, and at a rate of 16.5 frames per second for 160x120 images. Therefore, the method is applicable to the real-time processing. More over, since the Triclops library is currently optimized for thread parallel processing, a much greater speedup can be achieved on a dual-processor NT machine (we plan to move the system there in the near future).

The speedup is significant (between 1.70 and 2.15). Again, the speedup is defined as a ratio of an average non-ROI processing time per frame (488 ms) to the actual running time for this frame (with the new method). It is defined for all possible types of motion under any circumstances. However, the total amount of computation for stereo processing per frame (required for the Sum of Absolute Differences algorithm) is estimated as [22]: $N^2M^2d(C-1)P$, where $N^2$ is the image size, $C$ is the number of cameras (three for the system used), and $P$ is the number of operations per one square difference calculation. According to this, the speedup per frame should be proportional to the ratio between ROI and image areas. Experiments show that, for example, 6-8% ratio yields a gain of slightly more than 1.74 over non-ROI implementation. One of the reasons is that rectification (distortion removal) is still done on the entire image.

Another reason is that actual region sizes (for the correspondence matching between cameras) also include the number of disparities $d$ searched: for a K by L region, a left-to-right pass is done for a $K(L+d)$ region, and a top-to-bottom pass is performed for a $(K+d)L$ region. Matching on the entire image does not encounter this effect since there is obviously no data outside the image boundaries. Yet, these reasons do not account for the differences between the theoretical and experimental speedup. An implementation maximizing potential advantages of using ROIs can bridge this gap.

Motion trajectories for both hand movements in a 3-D space are shown in Figure 7. Zoom gesture trajectory is shown as a solid line, and translation is denoted with a dashed line. It is obvious that zoom gesture represents a significant change in $Z$ coordinate relative to the almost constant depth during the translation. Trajectory determines the type of gesture and intermediate hand positions. Data from trajectory computations will be transferred dynamically into an appropriate (for the task) interactive visualization software, where object of interest will repeat the same motion trajectory. Trajectory analysis is domain specific. Currently, we have HCI experts determining necessary gestures for controlling different visualization packages.

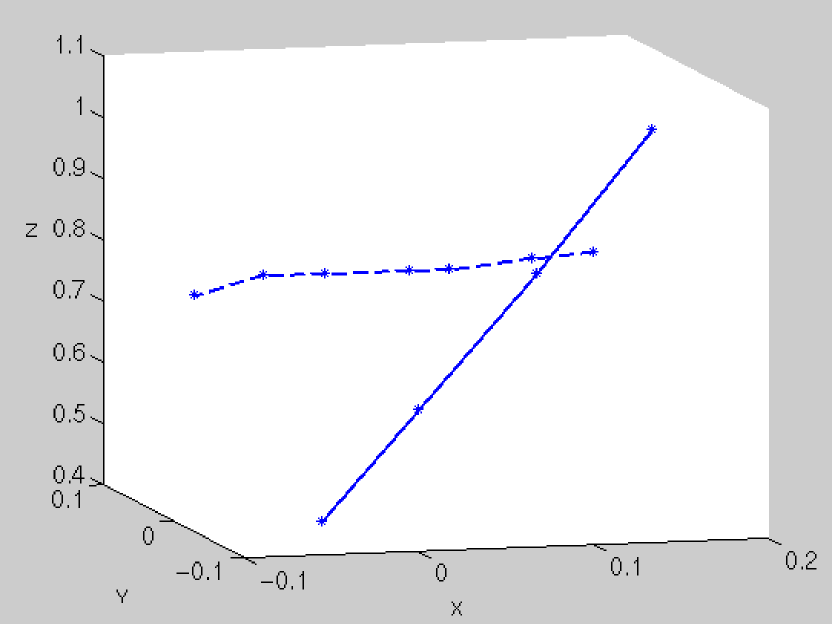


Figure 7: Motion trajectories for both hand movements in a 3-D space: zoom gesture trajectory is shown as a solid line, and translation is denoted with a dashed line.

Table 3 summarizes application of the method to a total of 800 images (20 sequences of 20 frames each and 10 sequences of 40 frames each) taken in different settings with one or two people performing various gestures. Quantitative analysis is performed for major groups of filters in the method (first column). The only exception is a combination of color feature filters and noise removal filters, because, depending on the scene complexity, total number of regions identified by color alone could reach hundreds (for example, separate parts of the hand, hairs, small background elements, and the other minor noise). The second column shows the number of detected regions identified as hands (at this step). This number slowly converges to the correct number of hands used for object manipulation as additional filters are applied. As a final result, all hands used for manipulation (where manipulation is identified by “grabbing”) were detected (a total of 1098 since some manipulations require two hands). At the end, 39 false positives (FPs) represent only 3.43%, false negatives are always less than 5%. This happens after all filter are applied, including the final depth/absolute size filter which demonstrates the benefit of having 3-D data available for moving regions of interest. The table also shows that static region detection yields not only a significant speedup, but also an improved localization of hands during manipulations. It produces the best overall improvement (decrease in the percentage of FPs between the current and the previous step). Hands are always detected in the initial frames (when they are present) which removes any unnecessary restrictions on interactors (as described in Section 1.2). It must be noted, however, that there is a restriction on the duration of a movement. Since range computation is selective, and at least three range images are needed, it was determined experimentally that the velocity of motion cannot be increased drastically. In fact, the duration of movements must be at least 1.42 seconds for 320x240 image size, and at least 0.91 seconds for 160x120 images.

Table 3: Results of hand(s) tracking in 800 images (20 sequences of 20 frames each and 10 sequences of 40 frames each).

| Filters applied | Detected | FPs | FPs, % |
|---|---|---|---|
| Color feature and noise removal | 10475 | 7026 | 67.07 |
| Region geometry | 3449 | 1987 | 57.61 |
| Static region detection | 1462 | 325 | 22.23 |
| Depth and absolute size | 1137 | 39 | 3.43 |

## 4 Additional Aspects of Real-Time Range

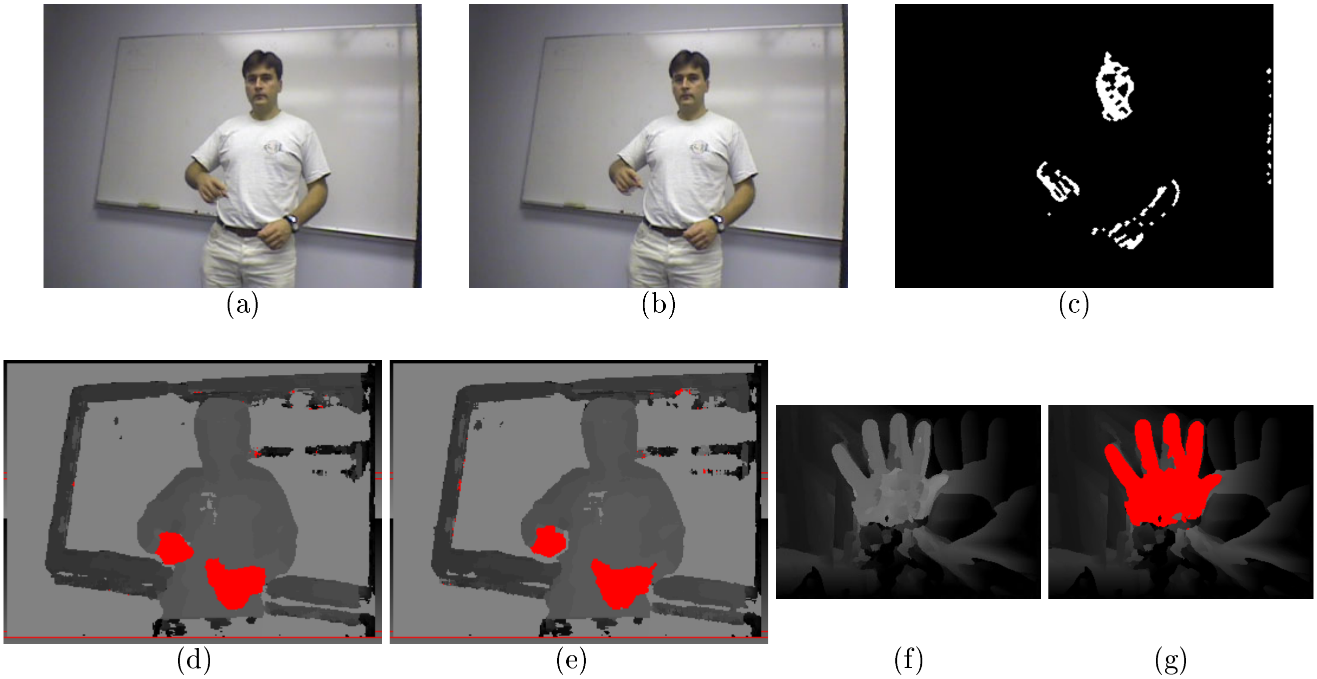
Increased processing speed can also facilitate a combination of intensity- and range-based input features. Range data enables localized search for specific features, which improves tracking reliability and speed.

Registered range data provides an additional information valuable for segmentation and tracking. Often, an object of interest can be separated from other objects or background by depth alone. In other cases, having fewer artifacts (that could complicate segmentation) in range information compared to intensity data is an important consideration for model matching [19].

Real-time constraints such as temporal correlation produce a possibility of searching within a smaller region, based on the match in the previous frame. For the range image, this involves depth planes immediately surrounding the plane where a hand (or face) was found in the previous frame. Subsequent search in the subset of the intensity data corresponding to these planes produces the position of the body part in the current frame. Therefore, intensity data is thresholded for the certain range and depth. Such combined use of input features produces not only a speedup due to a significant reduction in a search space, but also increased reliability due to a decreased number of false positives that could fall in such space. Rather than processing all pixels, this allows us to select only those pixels with the certain depth, based on the depth of the previously detected region of interest.

Two intensity images from a sequence of the speaking person are shown in Figures 8(a-b). More images are not displayed due to space restrictions. A skin detection algorithm is applied to the intensity data from Figure 8(a). Results of skin thresholding following color segmentation are shown in Figure 8(c). Pixels classified as skin are white. Note that, along with the face and hand information, it picks up up parts of other objects – a curtain on the right and a belt.

Instead of applying the color segmentation again, it is possible to take range data into consideration by selecting one or more depth levels where a region of interest was found (Figure 8(d)). Since the motion between two frames is small, the same level indicates approximate location of the hand in the next frame (red areas in Figure 8(e)). This level, along with the two closest depth levels (before and after), constitute the search space for the current frame (instead of the entire image). Search in the range domain prevented us from considering intensity-based segmentation artifacts (such as a curtain and a belt). Depth level search produces even better results when the hand is close to the camera (Figure 8(f-g)). Segmentations along intensity and depth channels also can be performed independently and then combined.


(a) (b) (c)

(d) (e) (f) (g)

Figure 8: (a-b) Intensity images of the speaker; (c) results of skin segmentation and thresholding; (d-e) range images with selected depth levels; (f-g) Depth level search results when the hand is close to the camera.

# 5 Conclusions and Future Work

This paper presented a new approach to a gesture-tracking system using real-time range on-demand. The system represents a gesture-controlled interface for interactive visual exploration of large data sets. The paper described a method performing range processing only when necessary and where necessary. This is achieved by a set of filters on the color, motion, and range data. The speedup achieved is between 1.70 and 2.15. The algorithm also includes a robust skin-color segmentation insensitive to illumination changes. Selective range processing results in dynamic regional range images (DRRIs). This development is also placed in a broader context of a biological visual system emulation, specifically redundancies and attention mechanisms.

The gesture-tracking system described in this paper will be responsible for supplying data manipulation parameters to interactive data exploration and collaborative visualization software. Processing one range image for every four color images is done at a rate of 10.6 frames per second for a 320x240 image size, and at a rate of 16.5 frames per second for 160x120 images. Therefore, the method is applicable to real-time processing. More over, as the Triclops library is currently optimized for thread parallel processing, a much greater speedup can be achieved on a dual-processor NT machine (we plan to move the system there in the near future).

Robustness of the approach is achieved with multiple input feature sets. Depth filters are necessary in addition to color, motion and shape filters. Increased speed of processing can also facilitate a combination of intensity- and range-based input features. Range data enables localized search for specific features, which improves tracking reliability and speed. Applications described are tolerant to temporary tracking confusions since the main goal of the gesture-controlled visualization is interactivity, not accuracy of intermediate states which are interpolated anyway DRRIs can be included in motion analysis, trajectory computation, and gesture recognition, to determine what types of gestures are natural and feasible for robust tracking and interpretation for interactive exploration of large data sets and virtual environments.

**Acknowledgments**

This work was done as a part of SAVAnTS scalable visualization project. I would like to thank the LLNL VIEWS Visualization project for Figures 1(a-b)); Mark Duchaineau, Sam Uselton, Randy Frank, and other members of Visualization Group of the Center for Applied Scientific Computing at the LLNL for their support and suggestions; Erick Cantu-Paz (CASC) for assistance in image acquisition; Alane Alchorn (CASC) for editorial review and recommendations; Regina Stadnik for the background production; Vladimir Tucakov and Don Murray at the Point Grey Research for technical support; Dmitry Goldgof at the Computer Science Department of the University of South Florida for invaluable guidance, and Min Shin at the Vision Lab (USF) for the face tracker.

## References

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):677–695, July 1997.

[2] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.

[3] M.-H. Yang and N. Ahuja. Recognizing hand gestures using motion trajectories. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, volume 1, pages 466–472, Fort Collins, CO, June 1999.

[4] J. M. Rehg and T. Kanade. Visual tracking of high DOF articulated structures: an application to human hand tracking. *Proc. of European Conference on Computer Vision*, 2:35–46, May 1994.

[5] J. J. Kuch and T.S. Huang. Model-based tracking of self-occluding articulated objects. In *Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration*, pages 666–671, Cambridge, MA, June 1995.

[6] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12):1325–1337, December 1997.

[7] Y. Yacoob and M. J. Black. Parameterized modeling and recognition of activities. *Journal of Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[8] R. Cutler and L. Davis. Real-time periodic motion detection, analysis, and applications. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, volume 2, pages 326–332, Fort Collins, CO, June 1999.

[9] D. J. Moore, I. A. Essa, and M. H. Hayes III. Exploiting human actions and object context for recognition tasks. In *Proc. of International Conference on Computer Vision*, pages 80–86, Corfu, Greece, September 1999.

[10] Y. Iwai, H. Shimizu, and M. Yachida. Real-time context-based gesture recognition using hmm and automaton. In *Proc. of International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 127–134, Corfu, Greece, September 1999.

[11] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. In *IEEE Nonrigid and Articulated Motion Workshop*, pages 90–102, San Juan, Puerto Rico, June 1997.

[12] S. E. Umbaugh. *Computer Vision and Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1998.

[13] M. W. Powell and R. Murphy. Position estimation of micro-rovers using a spherical coordinate transform color segmenter. In *Proc. of IEEE Workshop on Photometric Modeling for Computer Vision and Graphics*, pages 21–27, Fort Collins, CO, June 1999.

[14] J. Davis and M. Shah. Toward 3-D gesture recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(3):381–388, May 1999.

[15] A. Azarbayejani and A. Pentland. Real-time self-calibrating stereo person tracking using 3-D shape estimation from blob features. In *Proc. of International Conference on Pattern Recognition*, Vienna, August 1996.

[16] D. Gavrila and L. Davis. 3-D model-based tracking of humans in action: a multi-view approach. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, CA, June 1996.

[17] J. Segen and S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, volume 1, pages 479–485, Fort Collins, CO, June 1999.

[18] T. B. Moeslund and E. Granum. 3D human pose estimation using 2D-data and an alternative phase space representation. In *Proc. of IEEE Workshop on on Human Modeling, Analysis and Synthesis*, pages 26–33, Hilton Head Island, SC, June 2000.

[19] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Model-based force-driven nonrigid motion recovery from sequences of range images without point correspondences. *Image and Vision Computing Journal*, 17(14):997–1007, November 1999.

[20] L. V. Tsap, D. B. Goldgof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(5):526–543, May 2000.

[21] L. V. Tsap. Real-time local range on-demand for tracking gestures. In *Proc. of IEEE Workshop on on Human Modeling, Analysis and Synthesis*, pages 52–58, Hilton Head Island, SC, June 2000.

[22] K. Oda, M. Tanaka, A. Yoshida, H. Kano, and T. Kanade. A video-rate stereo machine and its application to virtual reality. In *Proc. of ISPRS '96*, 1999.

[23] J. W. Davis and A. F. Bobick. The representation and recognition of action using temporal templates. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, pages 928–934, San Juan, PR, June 1997.

[24] Point Grey Research Inc. *Triclops Stereo Vision System Version 2.1, User's guide and command reference.* Inc., Point Grey Research, Vancouver, BC, 1996.

[25] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition*, pages 196–202, June 1996.